

An Automated Template Approach for Generating Web-Based Conservation Planning Worksheets

Jeffrey J. Steiner,* Toshimi Minoura, and Mariko Imaeda

ABSTRACT

An automated web-form generator system was created to rapidly produce dynamic web-based inventory and assessment worksheets that are used by the USDA Natural Resources Conservation Service (NRCS) to describe the condition of natural resources on farms and ranches. These worksheets are used by NRCS as part of the criteria for creating a conservation plan to qualify landowners for payments under the USDA Farm Bill Conservation Title. Presently, most worksheets are filled out by hand or in personal computer spreadsheets. If worksheets such as these were available over the web, landowners could also self-assess the conditions on their farms or ranches before requiring assistance of NRCS conservation planners or technical service providers. Because many conservation worksheets have a similar style and behavior, we created an automated web-script generator that produces a dynamic web-page worksheet from a common template file and a configuration file. A worksheet-specific configuration file provides the details for the worksheet. The template, configuration files, and web scripts are all in PHP (PHP hypertext processor). Each configuration file required approximately 5 h to program, a savings of more than 4 d compared with a web-based worksheet programmed individually with a scripting language. In this way, various conservation worksheets can be quickly created for the web, as well as be easily maintained. As an example, we produced a suite of assessment worksheets used by Oregon NRCS with our automated web-form conservation worksheet system. Those worksheets are accessible at: http://yukon.een.orst.edu/ms_apps/wq_indexes/forms/index_menu.html (verified 5 May 2006).

THE web is becoming increasingly accessible to everyone and could provide a platform for landowner self-assessment and delivery of conservation planning and program services over the internet (R. Hatz, personal communication, 2004). Presently, NRCS conservation planners or technical service providers visit the farm or ranch to collect the information needed to complete a conservation plan for the landowner. The conservation plan is then used to determine eligibility for payments under the Conservation Title of the USDA Farm Bill. To expedite the planning process, it would be advantageous for landowners to contribute as much information as possible to the individuals who provide conservation plan-

ning services. This would reduce the total time needed by conservation planners to service a client, so each planner could service more clients. This is critical because of recent funding increases of Farm Bill conservation payments and greater accessibility of conservation programs to more farmers.

The NRCS uses various inventory and assessment worksheets as indicator tools to describe current, past, or future conditions of natural resources on farms and ranches (USDA-NRCS, 2000). Examples of these from the Oregon Water Quality Technical Note No. 1 include the nitrate index, phosphorus index, and nutrients, organics, and pathogens—livestock yard management worksheets (Table 1). Tools such as these can also be used to help educate landowners of water quality or other natural resources considerations, determine whether problems exist, set benchmark conditions, guide natural resources inventories, and understand how different conservation practices can reduce or eliminate pollution risks.

Presently, worksheets are available as printed pages or as spreadsheets on personal computers. Problematic to web delivery of worksheets is the great number of kinds of conservation indices and variations of worksheets specific for different states. A significant amount of time would be required to individually program each worksheet as a dynamic web-based worksheet in a scripting language (e.g., ASP Visual Basic [Microsoft, Redmond, WA], PHP [www.PHP.org], ASP.NET Visual Basic [Microsoft, Redmond, WA] or C# [Microsoft, Redmond, WA], JSP [Sun, Mountain View, CA], and perl [www.perl.org]).

Writing a typical web script for resource conservation is simpler than writing a program in an ordinary programming language, but still requires approximately 5 d of coding. Since most NRCS conservation worksheets have a similar style and operational function, we were able to implement a variety of worksheets with an automated worksheet web-form generator tool that encodes the common style and behavior of worksheets as a template. The specific content of each different worksheet is specified as a configuration file.

Another advantage of a template-based system is that a programming error need be fixed only once in the common template. Changes are then automatically propagated to all the worksheets that are produced from that template. Web-based delivery also allows updates to be made seamlessly without need for user updates.

This article is organized as follows: An overview of web-based worksheets, the N-index worksheet in particular, is given first. The details about implementa-

J.J. Steiner, USDA-ARS, National Forage Seed Production Research Center, 3450 S.W. Campus Way, Corvallis, OR 97331; and T. Minoura and M. Imaeda, College of Electrical Engineering and Computer Sci., Oregon State Univ., Corvallis, OR 97331. The use of trade, firm, or corporation names in this publication is for the information and convenience of the reader. Such use does not constitute an official endorsement or approval by the USDA or the Agricultural Research Service of any product or service to the exclusion of others that may be suitable. Received 12 May 2005. *Corresponding author (jjs@ars.usda.gov).

Published in *Agron. J.* 98:1181–1186 (2006).

Computers in Agriculture
doi:10.2134/agronj2005.0142

© American Society of Agronomy
677 S. Segoe Rd., Madison, WI 53711 USA

Abbreviations: N-index, nitrogen index; NRCS, United States Department of Agriculture, Natural Resources Conservation Service; PHP, PHP hypertext processor.

Table 1. The conservation worksheet tools presented in Oregon Water Quality Technical Note No. 1 (USDA-NRCS, 2000) that were implemented over the internet[†] using the automated worksheet web-form generator tool.

| |
|---|
| Pesticide indicator tools |
| Indicators for cropland, hayland, or pasture |
| Pesticide storage, handling, and disposal worksheet |
| Nutrient indicator tools |
| Nitrogen index |
| Western Oregon phosphorus index worksheet |
| Eastern Oregon phosphorus index worksheet |
| Nutrients—indicators for cropland, hayland, or pasture |
| Animal waste—indicators for pasture or range animals |
| Animal waste—indicators for totally or partially confined animals |
| Fertilizer storage and handling worksheet |
| Nutrients, organics and pathogens—livestock manure storage worksheet |
| Nutrients, organics and pathogens—livestock yard management worksheet |
| Salinity indicator tools |
| Indicators for receiving watercourses and water bodies |
| Indicators for flood and furrow irrigated areas |
| Sediment and turbidity indicator tools |
| Indicators for receiving watercourses and water bodies |
| Indicators for cropland, hayland, or pasture |

[†] Implementation over the internet is demonstrated at: http://yukon.een.orst.edu/ms_apps/wq_indexes/forms/index_menu.html (verified 5 May 2006).

tion of the automatic worksheet generation system are then described. Finally, some remarks about the system are given.

WEB-BASED WORKSHEETS OVERVIEW

Our automated web-form conservation worksheet system for creating web-based inventory and assessment worksheets was demonstrated by implementing 15 water quality indicator tools presented in Oregon Water Quality Technical Note No. 1 (USDA-NRCS, 2000) (Table 1). One of these, the Oregon Nitrogen-Index (N-index), is used to demonstrate our template-based process for implementing conservation worksheets as dynamic web pages. A slight modification to the Oregon NRCS N-index includes N loss to surface water by irrigation water-caused soil erosion (USDA-NRCS, 2002) (Fig. 1). The Portland NRCS State Office recommended these worksheets because they are commonly used by their staff. Only the specific code for the N-index is presented in this article.

| Source Factors | Factor Weight | None (0) | Low (1) | Medium (2) | High (4) | Very High (8) | Surface Loss Rating | Leaching Loss Rating |
|-------------------------------|---------------|---|--|---|--|--|----------------------|----------------------|
| Runoff Class | 1 | <input type="radio"/> Negligible & Very Low | <input type="radio"/> Low | <input checked="" type="radio"/> Medium | <input type="radio"/> High | <input type="radio"/> Very High | 2 | |
| Permeability Class | 1 | <input type="radio"/> Impermeable to Very Slow | <input type="radio"/> Slow to Moderately Slow | <input type="radio"/> Moderate | <input type="radio"/> Moderately Rapid to Rapid | <input checked="" type="radio"/> Very Rapid | | 8 |
| Annual Precipitation | 1 | <input type="radio"/> <10" | <input type="radio"/> 10"-15" | <input type="radio"/> 15"-30" | <input checked="" type="radio"/> 30"-45" | <input type="radio"/> >45" | 4 | 4 |
| Irrigation Water Management | 1 | <input type="radio"/> None | <input type="radio"/> Insignificant runoff and/or deep percolation - total amount applied = evapotranspiration + plant needs | <input type="radio"/> Little runoff and/or deep percolation - total amount applied does not exceed 110% of evapotranspiration + plant needs | <input checked="" type="radio"/> Some runoff and/or deep percolation - total amount applied does not exceed 125% of evapotranspiration + plant needs | <input type="radio"/> Significant runoff and/or deep percolation - total amount applied exceeds 125% of evapotranspiration + plant needs | 4 | 4 |
| Water Runoff Management | 1 | <input type="radio"/> Grasses, perennial vegetation CRP | <input type="radio"/> Residue management and/or strips/terraces plus buffers, filter strips or tailwater recovery | <input type="radio"/> Residue management and/or strips/terraces | <input type="radio"/> Conventional tillage, some residue | <input checked="" type="radio"/> Fallow condition, no residue | 8 | |
| Irrigation Erosion | 1 | <input type="radio"/> Not Irrigated or No Furrow Irrigation | <input type="radio"/> Tailwater recovery or QS>6 for very erodible soils or QS<10 for other soils | <input checked="" type="radio"/> QS>10 for erosion resistant soils | <input type="radio"/> QS>10 for erodible soils | <input type="radio"/> QS>6 for very erodible soils | 2 | 2 |
| Distance to Surface Water | 1 | <input type="radio"/> >500' | <input type="radio"/> 300' to 500' | <input checked="" type="radio"/> 100' to 299' | <input type="radio"/> 50' to 99' | <input type="radio"/> <50' | 2 | |
| Distance to Aquifer | 1 | <input type="radio"/> >100' | <input type="radio"/> 99' to 50' | <input type="radio"/> 20' to 49' | <input checked="" type="radio"/> 10' to 19' | <input type="radio"/> <10' | | 4 |
| Calculate | | | | | | | | |
| Total Weighted Ratings | | | | | | | 22 | 22 |
| Site Vulnerabilities | | | | | | | Low Potential | Low Potential |

Fig. 1. Example of a worksheet created by the automated web-form generator. Shown is the Oregon USDA-NRCS nitrogen index calculation with a modification using the calculation for the New Mexico USDA-NRCS surface irrigation erosion loss.

Web-Based Worksheet

In our approach, a single template is shared by different worksheets having similar organizational structure with the uniqueness of each realized by its configuration file. The template, configuration files, and web scripts are all coded in PHP (<http://www.php.net>; verified 5 May 2006), a widely used scripting language for creating dynamic web pages that can generate HTML (hyper text markup language) statements. Since PHP is a good tool for text manipulation, it can be used to generate a web script from the template and a configuration file. Once a web script is generated, it is loaded on a web server. Each time a user requests a specified conservation index calculation, the web script is executed by the server-side PHP interpreter.

Nitrogen-Index as an Example

The N-index worksheet is a matrix with 13 field characteristics and five value factors. Each field characteristic has a weighting factor. Each characteristic's weighting factor is multiplied by a value factor. The sum of the values gives the N-index result. Source and transport factors define the field characteristics. Source factors include N application rate, N form, commercial or organic fertilizer timing, and application method. Transport factors include runoff class, permeability class, annual precipitation amount, irrigation water management, water runoff management, irrigation erosion amount, distance to a surface water body, and distance to an aquifer. Surface and leaching N losses are both calculated by this indicator tool. The other 14 worksheets implemented by our system are similar to the N-index and also described in NRCS (2000). These worksheet examples can be accessed on the demonstration site to demonstrate the utility of our automated worksheet generation approach.

WORKSHEET IMPLEMENTATION

The N-index web-based worksheet (Fig. 1) was created following the template-based web script generation process (Fig. 2). The unique feature of our implementation method is that the PHP processor is used twice. In addition to being used to process web scripts on the web server, it is used to generate those web scripts from the template and configuration files. The online PHP processor on the server is activated each time a request for a worksheet is received from a web browser. On the other hand, the offline PHP processor is used only to prepare the web scripts. Once prepared, the web scripts can be activated repeatedly on the web server. The detailed steps are as follows:

1. The template file `indexgen.tmpl` (Fig. 3 and 4) is read by an offline PHP processor.
2. When the interpretation of the template is started, the configuration file `n_index.config` (Fig. 5) that specifies uniqueness characteristics of the N-index worksheet is read by the template. The template is an executable code that is interpreted by the offline PHP processor.

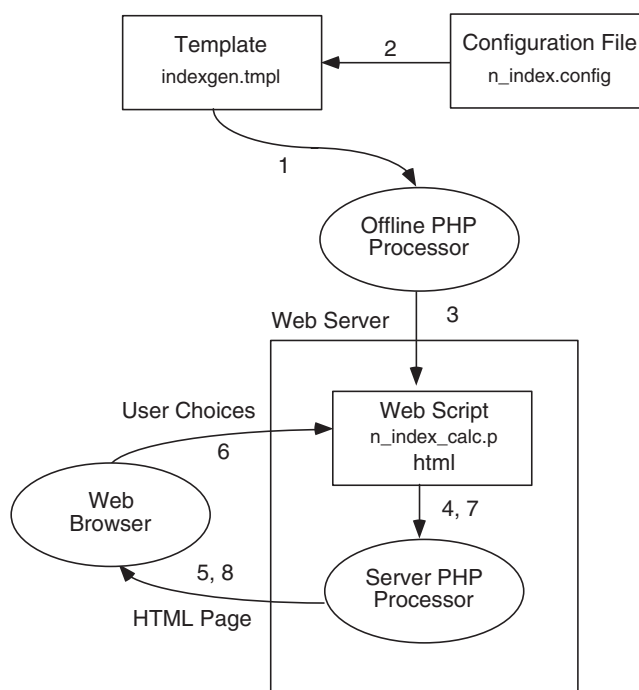


Fig. 2. The worksheet creation process by the automated template. Each process step is described in the text.

3. The web script (Fig. 6) is generated from the template and the configuration file by the offline PHP processor.
4. The PHP processor on the web server reads the PHP script. This PHP processor is different from the offline PHP processor.
5. The online PHP processor generates an HTML page (Fig. 7). A web browser receives the HTML page generated by the web script.
6. When the submit button on the worksheet is clicked, the user inputs are sent back to the web server.
7. The PHP processor interprets the same PHP script by using the user inputs and generates the HTML page including the calculation result.
8. The calculation result is sent back to the web browser.

Configuration File

A configuration file specifies the content of a worksheet by using the elements described in Table 2. The actual worksheet program can be generated from it without any human intervention. The configuration file `n_index.config` (Fig. 5) is for the N-index worksheet (Fig. 1). This configuration file makes the N-index worksheet unique from the others listed in Table 1. Each unique worksheet must have its own unique configuration file. The syntax of the configuration file is that of PHP. Lists of items are represented as arrays, some as associative arrays. An associative array stores each array element as a pair of a keyword and a value such as `label = > 'Surface Loss Rating'`. The PHP interpreter accepts this representation of data as PHP statements without any processing.

```

1  <?
2  for ($i = 0; $i < $n_factors; $i++) { // draw row for
3  each factor
4      $_label = $factors[$i]["label"];
5      $_weight = $factors[$i]["weight"];
6      $_level = 'level_' . $i;
7  ?>
8      <tr>
9          <td class="FieldCaptionTD">
10             <font class="FieldCaptionFONT">
11                 <?= $_label ?>
12             </font>
13         </td>
14         <% $_checked = $levels[<?= $i?>] == -1 ?
15         'checked' : '' %>
16         <td class="DataTD" colspan="1">
17             <font class="DataFONT">
18                 <%= $f_weights[<?= $i?>] %>
19                 <input type="radio" name="<?= $_level?>" value="-1"
20                 <%= $_checked %> style="visibility:hidden"/>
21             </font>
22         </td>
23     <?>
24         $descriptions = $factors[$i]['descriptions'];
25         $descriptions = $descriptions ? $descriptions :
26         $level_labels;
27         for ($j = 0; $j < $n_levels; $j++) { // for each level
28             $descript = $descriptions[$j];
29             ?>
30             <% $_checked = $levels[<?=
31             $i?>] == <?= $j?> ? 'checked' : '' %>
32             <td class="DataTD" colspan="1" valign="top">
33                 <input type="radio" name="<?= $_level?>" value="<?=
34                 <%= $_checked %>
35                 /><?= $_descript ?>
36             </td>
37         <?>
38         if (isset($subtotal_label)) { // single rating column,
39         show subtotal
40         ?>
41             <td class="DataTD" colspan="1">
42                 <%= $f_weights[<?= $i?>] * $level_weights[<?=
43                 $i?>] [$levels[<?= $i?>]] %>
44             </td>
45         <?>
46         } else { // multiple rating columns, show subtotal
47         foreach ($subtotals as $subtotal) {
48             $label = $subtotal['label'];
49             $hide = $subtotal['hide'];
50             if (in_array($i, $hide)) { // do not show
51                 ?>
52                 <td class="NoDataTD" colspan="1">
53                 </td>
54             } else { // show
55                 ?>
56                 <td class="DataTD" colspan="1">
57                     <%= $f_weights[<?= $i?>] * $level_weights[<?=
58                     $i?>] [$levels[<?= $i?>]] %>
59                 </td>
60             }
61         }
62     }
63     <?>
64     <% if ($errors[<?= $i ?>]) { %>
65         <td class="ErrorTD" colspan="1">*
66         </td>
67     <?>
68     <% } %>
69     </tr>
70 <?>

```

Fig. 3. A part in the template file indexgen.tmpl that generates the web script given in Fig. 6 according to the configuration file shown in Fig. 5. This part handles the processing of each factor in the worksheet.

Web Form

A web form is the final worksheet displayed by a web browser. This worksheet consists of HTML statements. The web form (Fig. 7) is for the N-index worksheet (Fig. 1). The data are displayed as a table specified by HTML statements, and the user interaction is handled by JavaScript statements. One web form is used to accept the inputs from the user and to display the calculation result. Figure 7 shows how the row for the Run-off Class factor in Fig. 1 is represented in HTML.

```

1  <? if (isset($subtotal_label)) { // single rating
2  column ?>
3      <td class="FieldCaptionTD" colspan="2">
4          <font class="FieldCaptionFONT">
5              Total Weighted Rating
6          </font>
7      </td>
8      <%
9          $total_rating = 0;
10         for ($i = 0; $i < $n_factors; $i++) {
11             $total_rating += $f_weights[$i] *
12             $level_weights[$i] [$levels[$i]];
13         }
14     %>
15     <td class="DataTD" colspan="1">
16         <font class="DataFONT">
17             <%= $total_rating %>
18         </font>
19     </td>
20 </tr>
21 <tr>
22     <td colspan="2" <%= $n_levels %></td>
23     <td class="FieldCaptionTD" colspan="2">
24         <font class="FieldCaptionFONT">
25             Site Vulnerability
26         </font>
27     </td>
28     <%
29         for ($i = 0; $i < count($range_upper_bounds); $i++) {
30             if ($total_rating <= $range_upper_bounds[$i]) break;
31         }
32     %>
33     <td class="DataTD" colspan="1">
34         <font class="DataFONT">
35             <%= $vulnerability_ranges[$i] %>
36         </font>
37     </td>
38 <?>
39 } else { // multiple rating columns
40     ?>
41     . . .
42     ?>

```

Fig. 4. The part of the template file indexgen.tmpl that generates the code for the index calculation part of the web script given in Fig. 6.

Web Script

A web script generates the initial worksheet, performs calculation, and generates the worksheet containing the calculated result. A web script is executed on a server, and the web forms generated are sent to a web browser. The web script n_index_calc.phtml (Fig. 6) written in PHP generates the N-index web form in HTML (Fig. 7). The statements inside a pair of <? and ?> are PHP statements that are executed (interpreted) by the online PHP processor.

Template File

The template file indexgen.tmpl (Fig. 3 and 4) specifies the style of a worksheet and its parameters. In the template code, some code is enclosed within <? and ?> or <% and %>. The code within <? and ?> is executed by the offline PHP processor when the web script is generated. The code within <% and %> is passed to the web script without any change, except <% is changed to <? and %> to ?> so that the code can be executed by the PHP interpreter on the web server. The rest of the code is also passed to the web script without any change.

The code used to generate the statements that process each factor (row) in the worksheet is shown in Fig. 3. Within the for loop, the following steps are taken for each factor:

1. Two variables are retrieved from the configuration file (Lines 3–4).

\$_label The label for the *i*th factor
 \$_weight The weight for the *i*th factor


```

1 <?
2 $worksheet_title = 'Nitrogen Index Worksheet';
3 $level_labels = array('None (0)', 'Low (1)', 'Medium
(2)',
4 'High (4)', 'Very High (8)');
5 $level_default_weights = array(0, 1, 2, 4, 8);
6 $subtotals = array(
7 array(label => 'Surface Loss Rating', hide => array(1,
7)),
8 array(label => 'Leaching Loss Rating', hide => array(0,
4, 6))
9 );
10 ?>
11 <%
12 $vulnerability_ranges = array(
13 array('Low Potential', 'Medium Potential', 'High
Potential',
14 'Very High Potential'),
15 array('Low Potential', 'Medium Potential', 'High
Potential',
16 'Very High Potential')
17 );
18 $range_upper_bounds = array(
19 array(15, 32, 66),
20 array(13, 29, 58)
21 );
22 %>
23 <?
24 $factors = array(
25 array(label => 'Runoff Class', factor_weight => 1.0,
26 descriptions => array(
27 'Negligible & Very Low', 'Low', 'Medium', 'High',
'Very High'
28 ),
29 level_weights => array(0, 1, 2, 4, 8)
30 ),
31 array(label => 'Permeability Class', factor_weight =>
1.0,
32 descriptions => array(
33 'Impermeable to Very Slow', 'Slow to Moderately
Slow',
34 'Moderate', 'Moderately Rapid to Rapid', 'Very
Rapid'
35 ),
36 level_weights => array(0, 1, 2, 4, 8)

```

Fig. 5. Configuration file *n_index.config* that makes the N-index worksheet unique.

2. Variable `$_level` indicates the level selected by the user (Line 5).
3. The label column is generated with `$_label`, and the weight column with `$_weight` (Lines 7–19).
4. To annotate the levels of the current factor, `$descriptions` is retrieved from the *i*th factor in `$factors` (Line 21).
5. The cells for the descriptions for the *i*th factor are filled (Lines 23–32).
6. The subtotal column(s) will be generated. If there is only one subtotal column defined in the configuration file, then one subtotal column is generated (Lines 33–39). Otherwise, multiple subtotal columns are generated. In this case, a cell is shaded if `$_hide` is true (Lines 40–57) and then the code for calculating the subtotals is generated (Lines 36–37, 52–53).
7. Finally, the code that displays an error flag is generated when an error is found for the *i*th factor (Lines 59–62).

The Calculate button displayed on the worksheet (Fig. 1) is used to submit a request to calculate the index rating. The steps that handle index calculation (Fig. 4) for one subtotal column are as follows:

1. Label total weighted rating is displayed (Lines 3–6).
2. The total weighted rating `$total_rating` is calculated as the sum of subtotal (Lines 8–11).

```

1 <?
2 $vulnerability_ranges = array(
3 array('Low Potential', 'Medium Potential', 'High
Potential',
4 'Very High Potential'),
5 );
6 $range_upper_bounds = array(
7 array(15, 32, 66),
8 );
9 <?php
10 include("../common.phtml"); // only get_param() is
used
11 $level_weights = array(); // 2-D array indexed by
[factor][level]
12 $total_ratings = array(); // totals of subtotal columns
13 $level_weights[0] = array(0, 1, 2, 4, 8);
14 . . .
15 $n_factors = 13; // # of factors
16 $n_levels = 5; // # of levels
17 $n_total_ratings = 2; // # of rating
columns
18 $f_weights = array(1, 1, 1, 1, 1, 1, 1, 1, 1.5, 1, 1,
1, 1);
19 . . .
20 <?
21 $total_rating = 0;
22 for ($i = 0; $i < $n_factors; $i++) {
23 $total_rating += $f_weights[$i] *
$level_weights[$i][$levels[$i]];
24 }
25 $total_ratings[0] = $total_rating;
26 ?>
27 . . .
28 <?
29 for ($k = 0; $k < 2; $k++) {
30 for ($m = 0; $m < count($range_upper_bounds[$k]);
$m++) {
31 if ($total_ratings[$k] <=
$range_upper_bounds[$k][$m]) break;
32 }
33 ?>
34 . . .
35 <?= $vulnerability_ranges[$k][$m]?>
36 . . .
37 <?
38 } // for each total rating
39 ?>

```

Fig. 6. The web script file *n_index_calc.phtml* used to generate the web form code shown in Fig. 7.

3. The result of Step 2 is displayed in a cell (Lines 13–16).
4. The label Site Vulnerability is displayed (Lines 19–23).
5. `$total_rating` is compared with each of the values in `$range_upper_bounds` to determine the *vulnerability range* (Lines 24–28).
6. The range is displayed in a cell (Lines 29–32).

REMARKS

We have used our automated web-form conservation worksheet generator to demonstrate rapid implementation of 15 Oregon NRCS water quality indicator tools as dynamic web forms that can easily be used by landowners, NRCS conservation planners, and technical service providers. We estimate the amount of time required to manually program each individual worksheet would be 5 d. Once our automated web-page generator was created, each of the worksheets required 5 h to produce. This was the time needed to write the configuration file for each specific conservation worksheet.

Our automated web-form generator provides a relatively simple and economic way to produce customized conservation index worksheets that can be readily

```

1<tr>
2  <td class="FieldCaptionTD">
3    <font class="FieldCaptionFONT">
4      Runoff Class    </font>
5    </td>
6  <td class="DataTD" colspan="1">
7    <font class="DataFONT">
8      1 <input type="radio" name="level_0" value=-1
9        checked style="visibility:hidden"/>
10     </font>
11  </td>
12 <td class="DataTD" colspan="1" valign="top">
13   <input type="radio" name="level_0" value=0 />Negligible &
14   Very Low
15 </td>
16 <td class="DataTD" colspan="1" valign="top">
17   <input type="radio" name="level_0" value=1 />Low
18 </td>
19 <td class="DataTD" colspan="1" valign="top">
20   <input type="radio" name="level_0" value=2 />Medium
21 </td>
22 <td class="DataTD" colspan="1" valign="top">
23   <input type="radio" name="level_0" value=3 />High
24 </td>
25 <td class="DataTD" colspan="1" valign="top">
26   <input type="radio" name="level_0" value=4 />Very High
27 </td>
28 <td class="DataTD" colspan="1"> 2 </td>
29 <td class="NoDataTD" colspan="1"> </td>
30 </tr>
31 <tr>
32   <td colspan="4">
33     <input type="submit" name='calc' value='Calculate' />
34   </td>
35   <td colspan="4">
36     <td colspan="4"></td>
37     <td colspan="4"></td>
38     <td colspan="4"></td>
39     <td colspan="4"></td>
40     <td colspan="4"></td>
41     <td colspan="4"></td>
42     <td colspan="4"></td>
43     <td colspan="4"></td>
44     <td colspan="4"></td>
45     <td colspan="4"></td>
46     <td colspan="4"></td>
47     <td colspan="4"></td>

```

Fig. 7. The web form HTML code sent to a web browser. The browser displays this code as the worksheet shown in Fig. 1.

accessed and used over the internet. Web-based worksheets are easy-to-use and should help increase land-owner participation in the conservation planning process and increase conservation awareness. It is estimated that with this initial implementation of Oregon conservation worksheets, NRCS conservation planning staff will save 200 h in time spent assisting with preparation of land-owner conservation plans (R. Hatz, personal communication, 2004). The web-based worksheet tools could also be adapted for use in a geographic information system (GIS) web database to organize information about fields within farms or ranches and help develop farm or ranch-wide conservation plans.

Table 2. Elements used in configuration file *n_index.config* (Fig. 6) that specifies the content of the N-index worksheet shown in Fig. 1.

| Variable name | Element |
|--------------------------------|---|
| \$worksheet_title | the title displayed at the top of the worksheet |
| \$level_labels | the labels for the columns |
| \$level_default_weights | the weight for each level if a particular factor requires different weights, those weights can be defined for that factor as level_weights in \$factors |
| \$subtotals | information on the subtotal columns each subtotal column includes label and hide the label of the subtotal column |
| label | the set of index numbers of the nonapplicable subtotal rows |
| hide | the labels for the ranking |
| \$vulnerability_ranges | the upper-bound of each range |
| \$range_upper_bounds | the upper-bound of last range does not need to be given. |
| \$factors | the information on each factor including label, factor_weight, descriptions, and level_weights |
| label | the label for the row |
| factor_weight | the weight for the factor |
| descriptions | the description for each cell of the factor |
| level_weights | the weight for each level of the factor |

The suite of conservation worksheets described in Water Quality Technical Note No. 1–Water Quality Indicator Tools (USDA-NRCS, 2000) is demonstrated at: http://yukon.een.orst.edu/ms_apps/wq_indexes/forms/index_menu.html (verified 5 May 2006). Additional documentation for the automated web-form generator conservation worksheet system is available free on request from the corresponding author. The authors ask that when any component parts of our system are used, that citation credit be given.

ACKNOWLEDGMENTS

This research was supported in part by a grant from the CALFED Bay-Delta Ecosystem Restoration Program (ERP 01-N25), in cooperation with P. Robins, executive director of the Yolo County Resource Conservation District. We also thank the Oregon USDA-NRCS state office for input about and testing of the conservation worksheet tools implemented. Particular thanks goes to Russ Hatz, leader for technology, and the staff of the USDA-NRCS state office in Portland, OR, for their guidance and assistance with testing of the web-based worksheets.

REFERENCES

- USDA-NRCS. 2000. Water quality technical note no. 1. Water quality indicator tools. USDA, Natural Resources Conservation Service, Portland OR.
- USDA-NRCS. 2002. Water quality technical note no. 10. Water quality indicator tools. USDA, Natural Resources Conservation Service, Albuquerque, NM.